

☐ Gr. 1, J.-P. Haslinger, MSc☒ Gr. 2, DI (FH) G. Horn, MScName Smola, ThomasAufwand in h Punkte Kurzzeichen Tutor / Übungsleiter / **1. Cäsar Verschlüsselung****(12 Punkte)**

Die Cäsar-Verschlüsselung (auch als Verschiebechiffre bezeichnet) ist ein einfaches Verschlüsselungsverfahren, das auf Substitution basiert. Bei der Verschlüsselung wird jeder Buchstabe des Klartexts auf einen Geheimtextbuchstaben abgebildet. Diese Abbildung ergibt sich, indem man die Zeichen um eine bestimmte Anzahl zyklisch verschiebt. Die Anzahl der verschobenen Zeichen bildet den Schlüssel, der für die gesamte Verschlüsselung unverändert bleibt.

Im Nachfolgenden Beispiel wurde als Schlüssel 3 gewählt, das heißt zum ASCII Wert des zu verschlüsselnden Buchstaben wird 3 addiert, um den verschlüsselten Buchstaben zu erhalten.

Klartext: A B C D E F G H...

Verschlüsselter Text: D E F G H I J K...

Für den Klartext HAGENBERG ergibt sich somit der verschlüsselte Text KDJHQEHUJ. Details zur Cäsar Verschlüsselung finden Sie natürlich auch wieder im Internet (z. B. unter <http://de.wikipedia.org/wiki/Caesar-Verschlüsselung>).

Entwickeln Sie ein Programm `caesar` welches Textdateien ver- und auch wieder entschlüsselt. Es soll dabei wie folgt direkt von der Kommandozeile gestartet werden können.

```
caesar [ -e | -d ] key inputFile [ outputFile ]
```

Die einzelnen Parameter haben dabei folgende Bedeutung:

- `-e` (optional, Standardwert) Die bei `inputFile` angegebene Datei wird verschlüsselt.
- `-d` (optional) Die unter `inputFile` angegebene Datei wird entschlüsselt.
- `key` Schlüssel, der zum Ver- oder Entschlüsseln verwendet wird.
- `inputFile` Name der Eingabedatei.
- `outputFile` (optional) Name der Ausgabedatei. Falls nicht angegeben, soll die Ausgabe auf die Standardkonsole erfolgen.

Achten Sie neben der korrekten Implementierung des Verschlüsselungsalgorithmus im Besonderen auch auf eine sorgfältige Auswertung der Kommandozeilenparameter. Testen Sie alle möglichen Varianten ausreichend.

2. Dokumente zensieren**(12 Punkte)**

Gesucht ist ein einfaches Softwareprogramm zur automatisierten Zensierung von Textdokumenten. Das Programm soll eine bestehende Textdatei einlesen, eine Reihe von „sensiblen“ Wörtern darin finden und „schwärzen“ (= durch Platzhalter derselben Länge bestehend aus dem Zeichen „X“ ersetzen) und das Ergebnis in einer neuen Textdatei speichern.

Damit das Programm vielseitig verwendet werden kann, sollen die zu schwärzenden Wörter selbst ebenfalls aus einer Textdatei eingelesen werden, in der pro Zeile genau ein zu zensierendes Wort erfasst ist.

Das Programm soll den Namen `sensor` tragen und wie folgt über die Kommandozeile aufgerufen werden können:

```
sensor <inputFile> <blackListFile> <outputFile>
```

Die einzelnen Parameter, keiner davon optional, haben dabei folgende Bedeutung:

`<inputFile>` Dateiname des zu zensierenden Dokuments.

`<blackListFile>` Datei mit den zu schwärzenden Wörtern.

`<outputFile>` Ausgabedatei für die zensierte Version des Dokuments.

Implementieren Sie das Programm, indem Sie zunächst die Liste aller zu schwärzenden Wörter aus der Datei einlesen und in einer geeigneten Datenstruktur ablegen. Lesen Sie danach die Eingabedatei Zeile für Zeile ein und schwärzen Sie in jeder Zeile alle zu zensierenden Wörter, bevor Sie die entsprechende Zeile in die Ausgabedatei schreiben.